Chapter 3

Optimal Lumber Production from Softwood Sawlogs

David Baar, Advanced Numerical Methods,davebaar@numericalmethods.com
Lou Hafer, SFU/Comp Sci,lou@cs.sfu.ca
David Kirkpatrick, UBC/Comp Sci,kirk@cs.ubc.ca
Ramesh Krishnamurti, SFU/Comp Sci, ramesh@cs.sfu.ca
Philip D. Loewen, UBC/Math, loew@math.ubc.ca
Michael Monagan, SFU/CECM, monagan@cecm.sfu.ca
Marc Paulhus, Calgary/Math, paulhusm@math.ucalgary.ca
Mihaela Radulescu, UBC/Math, mihaela@math.ubc.ca
written by
Philip Loewen

30 September 1997

1 Problem Statement

This problem deals with the maximization of the "value recovery number" for the processing of raw logs at sawmills in British Columbia. This quantity has the units of dollars per cubic metre, and is defined as the dollar value of usable lumber produced divided by the volume of raw material processed. Sawmills in the interior of British Columbia are of particular interest because the logs processed there are typically of small diameter and the current volume recovered is low, typically 45% to 55%. Each log sawn in a given mill can be given its own value recovery number, and clearly maximizing the aggregated criterion is equivalent to finding out how each individual log can be broken down into boards of the greatest total dollar value.

The first cut running the length of a log is of central importance: it defines a plane one can visualize as being rigidly attached to the log; all subsequent cuts must be either parallel or perpendicular to this plane. Determining the best first cut on each incoming log, which might have nonconvex cross-sections ("cats"), be bent ("swept"), or even have a nonplanar central axis ("corkscrew"), is a complicated optimization problem that must be solved in 3–30 seconds. This is the length of time that elapses as the log moves along a conveyor between a laser-operated surface-measurement station and the first rank of saws the log will encounter during the manufacturing process.

Equipment in the sawmill can shift the front and back ends of the log laterally, and then spin the log about an axis parallel to the direction in which the conveyor is moving, to provide a prescribed orientation relative to the vertical bandsaw blades that will make the first cut. The lateral positions of these blades can also be varied to control the widths of the boards this first cut produces. Typically there are two or four vertical blades; some mills also have two chipper heads on the outside of the blade assembly. For example, in a mill with two blades and two chippers, a log can be split into three planks with planar vertical sides. Figure 1 shows the end view of a possible first cut in such a mill: the two blades cut the lines separating regions (a) and (b), while the chipper heads reduce the material in region (c) to wood chips. As shown in Figure 1, the "centre cant" (region (a)), is typically wider than the "flitches" (region (b)). One reason for this is that the chain used in the log transport unit limits the separation of the two innermost blades to at most two or three inches, depending on the equipment.

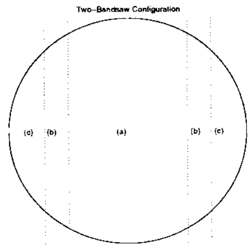


Figure 1: Two-bandsaw configuration.

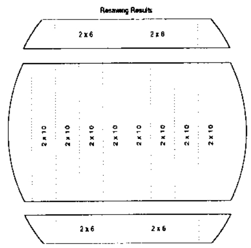


Figure 2: Sample resawing results.

After the first cuts are made, the three timbers in regions (a) and (b) are resawn to produce lumber of standard dimensions. All three are laid down on one of their cut surfaces and passed through ranks of either bandsaws or circular saws. For the flitches labelled (b), dimension lumber lying on its wide face results; for the centre cant (a), resawing produces dimension lumber standing on end. Figure 2 shows a possible breakdown of the boards cut from the log in Figure 1. (Note that the flitches shown at the top and bottom of Figure 2 and the centre cant between them are resawn on quite different pieces of machinery in an actual mill; also, there is no assertion here that the breakdown in Figure 2 is optimal.)

The dollar values of lumber vary according to market conditions, grade, and dimension. For the purposes of this discussion, we considered only the most important of these factors, namely, the correlation between the dimensions of a board and its wholesale value. J.E. Aune of MacMillan Bloedel supplied the typical values in Table 1 below. This table gives the price per cubic metre, calculated as the product of the base price with the length factor. Thus, a load of $1 \times 10''$ boards all 16 feet long is worth \$170 per cubic metre, whereas a load of 1×10's all 10 feet long is worth only $\$170 \times 0.70 = \$119$ per cubic metre. Notice that the length factors in this table are not simply increasing (multiples of 8 being especially valuable), and that differences in the length factors can reverse the trend suggested by the base prices. For example, 16-foot lengths of 2×6 are worth only $\$190 \times 0.80 = \$152$ per cubic metre, but ripping those boards in half lengthwise to produce 16-foot 2×3's yields a value of \$160 per cubic metre.

The problem first stated at the Workshop was to suggest efficient ways to maximize the value recovery number defined above for each given log. We were given very few constraints within which to work, the most important being that every cut had to run the length of the log, and that all cuts

Nominal Size	Base Price ($\$/m^3$)	Length Factors						
		8′	10′	12′	14′	16′	18′	20′
1 × 3″	\$100	0.60	0.70	0.80	0.90	1.00	0.90	0.85
1 × 4″	\$120	0.60	0.70	0.80	0.90	1.00	0.90	0.85
1 × 6″	\$150	0.60	0.70	0.80	0.90	1.00	0.90	0.85
1 × 8″	\$150	0.60	0.70	0.80	0.90	1.00	0.90	0.85
1 × 10″	\$170	0.60	0.70	0.80	0.90	1.00	0.90	0.85
2 × 3″	\$160	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2 × 4″	\$180	1.00	0.70	0.80	0.90	1.00	0.90	1.00
2 × 6″	\$190	0.80	0.80	0.80	0.80	0.80	1.00	1.00
2 × 8″	\$210	1.00	0.90	0.80	0.90	1.00	0.90	1.00
2 × 10″	\$240	0.90	0.80	1.00	1.00	0.90	0.90	0.80

Table 1: Typical Lumber Value = Base Price × Length Factor

after the first had to run either parallel to or perpendicular to the plane of the first cut. (The written specifications of the problem do not place restrictions on the number of parallel cuts that can be made in a single pass, although practical limits were discussed in meetings between team members and MacMillan-Bloedel's representative J. E. Aune.) We were allowed to move the log laterally at either end, and rotate it about an axis parallel to the conveyer. An improvement of 1% in value recovered would be valuable to MacMillan Bloedel. Later, we learned that MacMillan Bloedel were primarily interested in finding efficient ways to determine the best first cut. The sequence here is immaterial: *both problems amount to the same thing.* It is clear that regardless of what the first cut may be, once it has been made, the products that result should be resawn in the best possible way in order to maximize profit. So if the goal is to make an intelligent choice of the first cut, the preference of one choice over another can only be determined by optimizing the products resulting from both choices, and associating their payoff values with the corresponding initial decisions. This is why there is no sure way to determine the best first cut without going through some procedure that determines the entire bundle of products that subsequent steps will ultimately associate with the first cut.

If we relax the problem from "find the best first cut" to "find a very good first cut", a second interpretation becomes possible. Experience may well lead to effective "rules of thumb" that often result in first cuts that are very close to optimal, despite being comparatively cheap to implement. In order to gain this experience, of course, one must completely solve the log breakdown problem in many individual cases. But, if we assume that this has been done and that the results are known, there may be some way to make intelligent use of this accumulated wisdom to save computation time, produce better results than the necessarily approximate optimization methods now employed, or even do both.

Here are succinct statements of our two problems.

Problem 1. *Given a log of known dimensions, determine the best location for the opening cut. As a byproduct, completely describe all saleable products that can be produced following this first cut.*

Problem 2. *Given extensive log-optimizing experience, propose new methods to determine a nearly-optimal first cut, without necessarily generating any more information about subsequent stages of processing.*

Problem 1 is a mathematical optimization problem, in which there are three continuous variables (lateral shift of the butt end, lateral shift of the top end, and rotation angle) and a number of discrete variables (number, relative positions, and dimensions of each board in the bundle to be manufactured). Various constraints also come in, due to the number of sawing stations available to process incoming timber, the number of blades at each station, and the physical limits on their relative positions. These features typically vary from one mill to the next, and one expects the optimal breakdown to vary along with them.[1] What is more, any deterministic representation will fail to capture the random errors introduced by the equipment in an actual mill. (For example, Jan Brdicko says that ordering the rotation of a log by some angle θ about its long axis will trigger a sequence of mechanical events that rotate the log by an actual angle $\tilde{\theta}$ that may differ from θ by 10 or 15 degrees.) Thus an implementable solution should take into account the need for robustness with respect to perturbations of the specifications—the truly practical problem might be to maximize the *average* value recovery under plant uncertainty, or even to maximize the *worst-case* value recovery over parameter regions near to the values of top shift, translation, and rotation that our methods propose.

For our first foray into this field, we ignored all these complicating factors, and dealt only with the single constraint explicitly stated in the written problem statement provided to all workshop participants—namely, that all cuts must be parallel or perpendicular to the first one made[2].

1 Input Data

The information on which the optimization described above is to be based comes from a laser scanning station positioned along the conveyor that moves the log toward the saws. If we consider the z-axis to lie along the direction in which the conveyor is moving, the scanner records the (x, y)-coordinates of 36 points around the perimeter of the log every time its z-coordinate increases by some fixed amount. MacMillan Bloedel provided 15 data sets of such coordinates, which we reformatted into a collection of 45 files—3 for each log. In these files, all measurements are in inches. File `x07.dat` contains a $36 \times N$ matrix of x-values in ascii format; files `y07.dat` and `z07.dat` contain $36 \times N$ matrices with the corresponding y and z-values. Taking the k-th column of these three matrices gives the 36 data values for the log in position k: the number of columns (N) varies, depending on the length of the log. The naming conventions are simple: log numbers from `01` to `15` can take the place of `07` in the description above. For each of these logs, the fixed spacing between z-coordinates of successive measurements is 4 inches, so the z-data is particularly simple. (Each column in matrix `znn.dat` is constant, and the values are simply $0, 4, 8, 12, \ldots$.) These data sets are available for anonymous ftp, from `ftp.math.ubc.ca`, in directory `/pub/loew`.

The test data provided by MacMillan Bloedel can be visualized in several ways. A three-dimensional mesh plot joining the measured data points helps one to visualize the log as a whole (see Figure 3 below). In each of the data sets provided, this picture looks very much like a truncated right circular cone. In many cases the lateral sides of the cone are very nearly parallel to its axis, so that the truncated cone is virtually a cylinder. In other cases, the taper is pronounced. (J. E.

[1] Comparing optimal breakdowns among various mill configurations could thus provide valuable guidance in choosing the configuration for a new or reconditioned sawmill.

[2] Some of our formulations may be adaptable to accommodate the curved cuts now being proposed in the lumber industry.

Aune writes, "Log scalers talk of taper as the length of log corresponding to a one inch reduction in diameter, i.e., one inch in twelve feet meaning little taper, one inch in six being a lot of taper.")

Another way to plot the data is to make a two-dimensional plot showing the rings of measurement points superimposed on one another. This view, which amounts to looking straight down the log along the conveyor axis, reveals the effect of taper most clearly. It can also reveal situations where the natural axis of the log does not line up with the conveyor axis. (See Figure 5 below.)

To see the taper, one can plot the distance from each of the measured points on the log to either the conveyor axis or to (some estimate of) the log's natural axis as a function of distance along the log. (See Figure 4 below.) When the conveyor axis and the natural axis coincide, this provides a series of vertical clusters of 36 radius values: the taper can be estimated by fitting a straight line to this data, but (as J. E. Aune has noted) a parabolic fit is usually better.

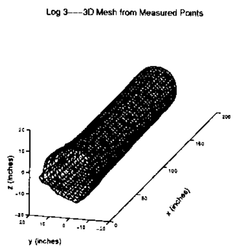


Figure 3: 3D Image of Test Log No. 3.

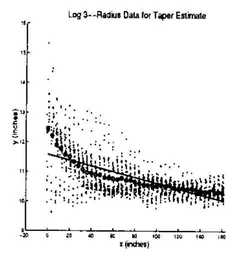


Figure 4: Radius data for taper estimate.

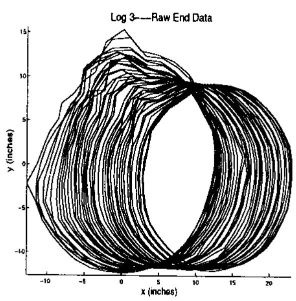


Figure 5: Raw end data.

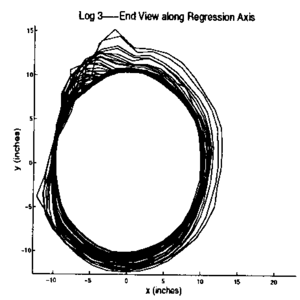


Figure 6: End view along regression axis.

A summary of physical characteristics of the test logs appears in Table 2. Here, the natural axis of the log was computed using linear regression through the set of all input data points, the taper was estimated as above, and the column labelled "top shift" gives the minimum distance the log's top needs to be moved to make the log's natural axis parallel to the conveyor axis. The table shows that log number 12 tapers very little, while log number 13 tapers a lot. Most of the logs landed on the conveyor within a fraction of a degree of the conveyor axis, but perfect alignment could still entail shifting one end by a noticeable amount; the alignment is worst for log number 3, where a shift of over 13 inches seems to be indicated. We emphasize that this is actually the relative shift between the top and bottom ends of the log: in a working mill, it may also be necessary to shift the whole

Log Number	Length L (inches)	Log-to-Conveyor angle θ (degrees)	Top shift (inches)	$\triangle \text{diam}/\triangle z = 1/x$, where $x = \ldots$
1	168	0.96	2.8	42
2	248	0.47	2.0	229
3	160	4.67	13.1	52
4	160	2.77	7.7	98
5	160	1.59	4.4	108
6	160	1.11	3.1	78
7	160	0.40	1.1	312
8	160	0.44	1.2	198
9	160	0.96	2.7	37
10	160	1.96	5.5	56
11	160	0.82	2.3	45
12	156	1.86	5.1	645
13	160	1.26	3.5	36
14	124	0.81	1.8	145
15	1100	0.12	2.3	104

Table 2: Physical Characteristics of Test Logs

log parallel to the cutting axis so that it is roughly centred as it approaches the headrig. For the purposes of optimal breakdown, however, we can ignore this rigid translation simply by assuming that there are no limits on the lateral positions of the saws.

(The top shift amount equals $L \tan\theta$, where L is the length of the log's projection onto the conveyor axis and θ is the angle between the log's natural axis and the conveyor axis, as shown in Table 2. It is included mainly to convey a sense of scale, since the required motion may not lie in either a horizontal or a vertical plane, and when it has been completed, the log's natural axis may be some (parallel) distance away from the axis of the conveyor.)

Figures 3–5 illustrate the various properties of test log number 3. This log has the highest value we found for the angle between the natural axis and the conveyor axis (and consequently for the top shift amount), and tapers more than one inch in five feet.

2 Mathematical Optimization

With regard to the full optimization problem of most efficient log breakdown, we have progress to report on several fronts: selecting the cutting axis, breakdown for cylindrical logs, and cutting planks from flitches. Although a week was not enough time to produce a full three-dimensional optimization package, we are confident that the analysis and ultimate solution of these subproblems will provide insight and perhaps even auxiliary code that will be useful in the general value-recovery problem.

1 Axis Selection by Standard Means

The regression axis used to investigate the given logs in the previous section is optimal in one sense: it is the line for which the sum of all squared distances to the data points (measured in planes perpendicular to the conveyor axis) is the least possible. It clearly gives a better approximation to

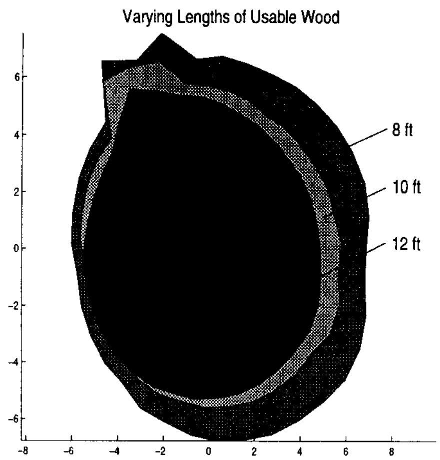


Figure 7: Computing the volume of usable wood.

a reasonable cutting axis than the conveyor axis does, and has the advantage of being very easy to compute. An alternative is to choose a line that minimizes the sum of squared distances to the data points (measured not in planes, but in $\mathbb{R}^3$): this can be determined using the Singular Value Decomposition at somewhat greater computational expense. Our experience on the test data described above was that the SVD approach gave axes that were indistinguishable from the regression lines for practical purposes, so we will not discuss SVD in detail here. However, for logs with substantial curvature, the SVD may be an appropriate tool to identify the natural plane in which the log lies. (The results for logs in our data set were inconclusive, since they exhibit very little "sweep".)

2 Axis Selection based on Maximum Usable Volume

The standard mathematical techniques just described are optimization-based, but they involve criteria that are geometric, and not easily related to the real objective of maximum value (or volume) recovery. Another way to choose a cutting axis without going through a complete breakdown optimization is to choose the axis along which you find the maximum volume of "usable wood". By usable wood, we mean wood of allowable lengths from Table 1, i.e., $8'$, $10'$, $\ldots$, $20'$.

To find this, postulate a cutting axis and have the computer construct a contour map of the butt end of the log by shading all areas that go $20'$ down the log, $18'$ down the log, $16'$ down the log, and so on, down to $8'$. See Figure 7. Multiply the shaded areas by their associated lengths to find the volume of usable wood associated with the given cutting axis. Use a continuous optimization routine to choose the cutting axis for which this volume is maximized.

A randomized optimization approach would be easy to implement. Given a proposed cutting axis, Marc Paulhus proposes that one choose at random a pivot point and a direction to move the end of the log, and test if this perturbation improves the usable volume. If so, then accept it; otherwise, keep the original orientation. Repeat this process until random choices of a given size scale no longer help. At this point reduce the size of random perturbations and continue.

In Paulhus's experience, this "tweaking process" will quickly find the maximum weighted contour map that corresponds to the cutting axis which provides the greatest potentially usable volume. Even if this is not the axis you wish to cut, it might be a good starting point for a search.

3 Vertical Slices for Cylindrical Logs

When we looked at the wire frame model of the first log from the test data MacMillan Bloedel supplied (compare Figure 3), we were struck by the log's almost perfect cylindrical shape. We had been discussing an algorithm for determining the axis to cut along by finding the largest cylinder that fits entirely inside a given log, expecting that we would later have to do better even than this. When we realized that our suite of test data included a number of logs that are very nearly cylindrical, we put this discussion aside in order to look directly at the problem of breaking down a cylindrical log. We wanted to determine whether the "centre cant plus side flitches" approach reviewed by MacMillan Bloedel was the best sawing strategy after all. The "cant" algorithm is described below.

1. Find the largest width w in the list of usable dimensions (see Table 1) that fits horizontally into the log. If the log radius is r and the prescribed width is w, one can make a rectangular plank of any thickness up to $t^* = \sqrt{4r^2 - w^2}$ down the length of the log.

2. Stack as many planks of the chosen width w as possible vertically into the log. Since planks come in thicknesses of 1 and 2 inches, we can pack any integer height. Thus the thickness actually to be used should be

$$t = \left\lfloor \sqrt{4r^2 - w^2} \right\rfloor .$$

   This thickness will be cut into two-inch thick planks in preference to one-inch thick ones as the former are (always?) more valuable. So there will be $t/2$ two-inch planks if t is even, and $(t-1)/2$ two-inch planks plus one one-inch plank if t is odd.

These two steps determine the width and thickness of the centre cant. MacMillan Bloedel argued that this should be a good approach because it tries to extract as many of the most valuable planks from the log as possible. Jan Aune also pointed out two optimization-based refinements to this method:

1. The number of planks in the cant should be optimized over different rotations of the log. This is irrelevant for a cylindrical log (or, more generally, for a truncated cone), but will often be important for a log of irregular shape.

2. Since the width w and thickness t of the centre cant must be chosen from a short list of marketable values, there will often be freedom to shift the cant to the left, to the right, up, or down inside the log. For example, moving the cant slightly to the left may permit a thicker plank to fit on the right—or vice versa.

Macmillan Bloedel stated that they did not know whether this centre cant approach was the best in general or not. We believe that it is not, and describe a better general approach below.

An obvious problem with the cant approach is that the four pieces which are left over are often not efficiently utilized. We thought that it might be better to split the log vertically into slices, preferably two inch thick slices, and then optimally pack each vertical slice. The intuition is that this approach will utilize the top and bottom of the log better than the cant approach because more often it will be filled with the thicker planks. Moreover, this approach still yields a simple cutting solution—there is one vertical cutting pass required for the log followed by one horizontal cutting pass for each vertical slice.

To test our hypothesis that the vertical slice approach is better than the cant approach, we constructed the following experiment. We implemented a program which computes a packing for the cant approach as described above where we allow it to add any number of planks to the left of the cant (and symmetrically to the right), and above the cant (and symmetrically below it) to fill up the remaining space. Then we implemented the vertical slice method with three variations:

- choose one 2″ wide plank centered in the middle of the log,
- choose one 1″ wide plank centered in the middle of the log, and
- choose two 2″ wide planks separated by the centre line of the log.

We computed the best packing for each and take the maximum of the three.

We used the sample market price data provided by MacMillan Bloedel to determine which was the optimum decomposition of the vertical planks.

Using only cylindrical logs, we applied both algorithms to a series of 101 logs of fixed length whose diameters ran from 10 inches up to 20 inches in 0.1-inch increments and took the average improvement in the wholesale value of the products introduced by our approach over the cant method. We repeated this experiment seven times, once for a log of each of the standard lengths. We did not make any cuts perpendicular to the long axis of the log (such as cutting a 20-foot board into a pair of 10-foot boards).

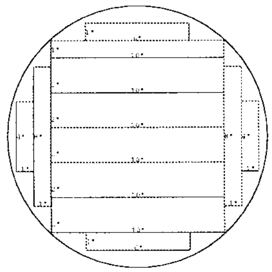
Some of the cutting schemes resulting from our implementation appear in Figures 8 and 9. The experimental results are summarized in Table 3. This table shows seven standard lengths, and the average improvement in value recovery for the slice algorithm over the cant approach. In most of the individual trials contributing to this average statistic, the slice algorithm gave a packing with the better dollar value. The cant approach did come out ahead in a reasonable number of trials, however, and occasionally the two methods tied. Obviously, one could try both methods and use the one with the greatest value.

Log/Lumber Length (ft)	Average Gain in Value using Vertical Slices (%)	Vertical Slicing over Cant Method		
		# Wins	# Losses	# Ties
8	10.8	97	0	4
10	2.8	69	29	3
12	0.6	52	46	3
14	5.1	77	20	4
16	2.1	67	23	11
18	2.0	64	34	3
20	11.8	97	1	3

Table 3: Average Improvement using Vertical Slices

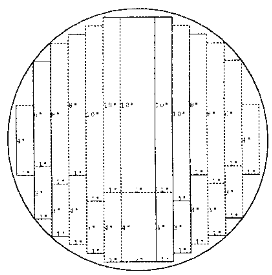
The reason for the high variance is due in part to the prices. The prices in Table 1 show that for 8′ and 20′ planks, the value of the 2″ thick timber compared to the 1″ thick timber is higher than that for the other lengths. If one compares the packings visually one can see clearly what is going on, namely that the vertical slice algorithm packs in more of the thicker lumber. Hence the higher prices for 8′ and 20′ lengths yield the highest improvements in Table 3.

Diameter 15" gives 82.6% yield of 20.28 cubic feet for $98.44



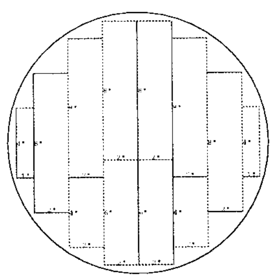
(a) Value-Optimized Cant Breakdown

Diameter 15" gives 87.1% yield of 21.39 cubic feet for $81.22



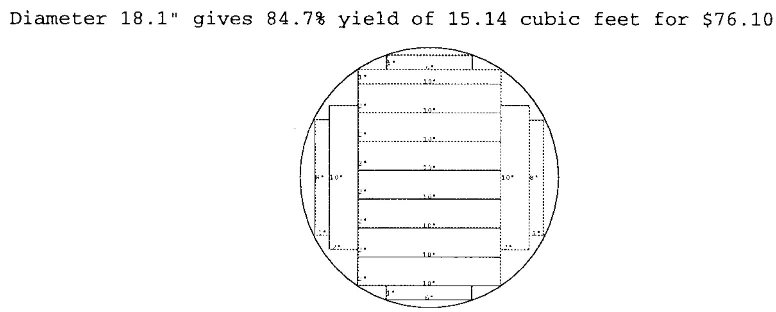
(b) Volume-Optimized Slice Breakdown

Diameter 15" gives 81.5% yield of 20 cubic feet for $111.8

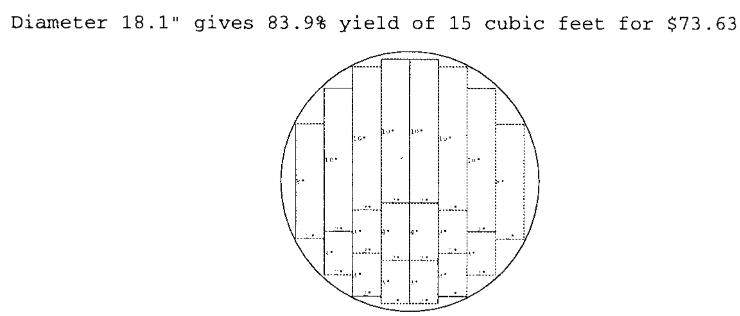


(c) Value-Optimized Slice Breakdown

Figure 8: Possible Breakdowns of a Cylinder with Diameter 15″, Length 20′



(a) Value-Optimized Cant Breakdown



(b) Value-Optimized Slice Breakdown

Figure 9: Possible Breakdowns of a Cylinder with Diameter 18.1″, Length 10′

We have not yet implemented the cant-placement optimization suggested by MacMillan Bloedel. Of course, this same idea should be applied to both algorithms. But, because the vertical slice algorithm packs the top and bottom better already, it will not benefit from this refinement as much as the cant algorithm. Thus the percentage improvements reported here are somewhat higher than we would expect in practice. Other optimizations may further narrow the gap.

Before we attempted to implement this optimization we found out from Jan Brdicko of MPM Engineering that the vertical slice approach, called "live sawing" in the industry, is already known, and indeed known to be more efficient in general. "So why isn't it used?" we asked. We were informed that in some (very few?) mills, it is in fact used, but because most(?) saw mills use only 4 vertical blades (some have 6) it cannot be done in a single pass. However, if the vertical slice algorithm is significantly better, then one could imagine making two passes. Given this very positive data, MacMillan Bloedel might consider a study to implement and compare a vertical slice algorithm with the cant algorithm (with optimizations included) on real data to determine how much better it would be in practice, and whether it would be worthwhile to reconfigure any sawmills to accommodate the vertical slice approach.

4 Optimized Flitch Sawing

To make efficient use of the sideboards labelled (b) in Figure 1, it is necessary to know how to get the maximum possible amount of lumber out of a one- or two-inch thick plank with irregular edges. This is a stand-alone sawmill optimization problem of independent interest, but it also plays a vital role in the full problem, since a complete analysis of the full-log problem requires that the optimal value of each cutting product be worked into the calculations. Also, the pieces of timber produced by the live-sawing method just discussed are mathematically equivalent to flitches. Finally, the flitch-sawing problem is a natural place to start for mathematical reasons: it is (approximately) two-dimensional instead of three, so it should be easier to solve than the full problem; furthermore, ideas arising in the 2D case could possibly turn out to remain useful in the general case.

We imagine the sideboard lying on one of its cut faces, and assume for simplicity that all of its edges are vertical. Then the measurement points of the edges of the sideboard provide the vertices of a polygon F in the (x, y)-plane, from which we desire to cut rectangular subsets whose total dollar value is a maximum. The long sides of all these rectangles must be parallel. We have partial results for two approaches, both of which apply to an arbitrary plane polygon F (although we expect F to be long and narrow): first, a method for maximizing the length of a board with given width lying inside F; second, a simple way to detect if a board of given width fits inside F.

(a) Maximum Length for Given Width

Given the vertices of closed polygon ("flitch") F in $\mathbb{R}^2$ whose sides do not intersect, one can easily compute the intersection between F and a horizontal strip $S(a, b) = \{(x, y) : a \leq y \leq b\}$, where a and b are prescribed saw positions and $w = b - a$ is one of the widths of standard lumber. The idea is simply to work around the perimeter of the flitch F, taking note of the points where line segments between successive vertices (measured data points) cross the top or bottom of the strip. The polygon of intersection, corresponding to the shape of the board that would remain after sawing along the lines $y = a$ and $y = b$, has for vertices the collection of all the crossing points together with all the vertices lying inside the strip. With this description for the set $C = F \cap S(a, b)$ in hand, finding the maximum number of rectangles ("planks") in the set C and their lengths is also straightforward. To illustrate, Figure 10(a) shows a 12-foot long flitch and two parallel lines separated by 4 inches. In Figure 10(b), the computation of yield for this scheme has been determined by the method above.

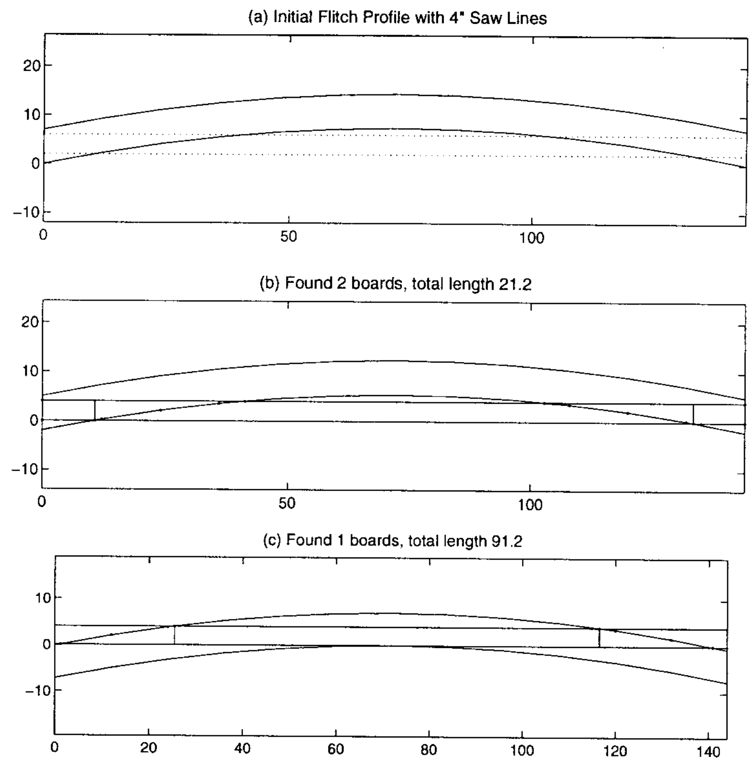


Figure 10: Steps in maximizing length for given width.

When these capabilities are combined with a method for shifting and rotating the flitch F into an arbitrary location relative to the saw lines $y = a$ and $y = b$, one has a method for evaluating the possible number and length of boards of a specific cross-section associated with a particular cutting scheme. Optimizing the value of such a solution over the possible shifts and rotations of the given flitch can be done by standard subroutines. Figure 10(c) illustrates the outcome for this example.

For wide flitches, it may be possible to extract more than one usable board. Our methods for finding the length of boards compatible with a given pair of cut lines extend easily to finding the lengths of a bundle of boards lying side-by-side, in various positions down the flitch. We would plan to use some discrete optimization engine to generate sequences of board widths for processing by this length-optimization algorithm, and do a rapid search for the best possible collection of widths (with its corresponding orientation).

One aspect of this method requiring further study is the objective function in the optimization just demonstrated. So far, we just maximize the total length of all boards produced. Thus, our implementation assigns the same value to one twelve-foot board as it does to four three-foot boards. This is obviously unrealistic, but in order to get reasonable results from any continuous optimization package, one needs an objective function whose gradient (exists and) is nonzero at the vast majority of points. The true objective function—valued in dollars—is constant in the whole range of lengths from 8 to 10, then constant again from 10 to 12, and so on. Further study would be required to find an objective function that captures the essence of the problem's discreteness while having no flat spots.

(b) Discrete Search Plus Oracle

Reversing the design in the previous subsection holds considerable potential. Under this scheme, the discrete optimization routine would propose not just a list of widths, but the full shape of a "template" of lumber, including lengths and relative positions of the finished boards. Then a continuous subprogram ("oracle") would be used to determine if the current template can actually be cut, and if so, to give the flitch orientation relative to the blades that makes it happen. The discrete driver would take the usual branch-and-bound or knapsack approach, although the details of how to structure the search space will take some careful thinking to work out. We can, however, describe an oracle that detects the viability of a given template.

We outline the special case in which the template to be realized is a single plank, 6 inches wide. Deciding whether or not this fits into a given flitch is more complicated than simply asking if the flitch is at least six inches wide everywhere along its length—the flitch in Figure 10 is seven inches across everywhere, and we did not even get eight feet of 4-inch wide lumber! The key concept here is convexity: a simple plank can only fail to be realizable if one of the vertices of the given polygon lies inside it. So if we can lay the bottom edge of the plank along a line through the flitch that lies above all the lower vertices and at least 6 inches away from all the upper vertices, the plank will fit inside the flitch. The line we want must therefore separate the set of lower vertices from a system of disks with radius 6 centred on the upper vertices. And a line has this separation property if and only if it separates the upper convex envelope of the flitch's bottom vertices from the lower convex envelope of the disk-augmented upper vertices. Figures 11 and 12 illustrate this: both involve computer-generated flitches with considerable taper and exaggerated random variations in upper and lower surfaces. Both figures show (i) the flitch profile itself; (ii) the upper vertices enlarged into disks of radius 6″; and (iii) the convex envelopes just mentioned. In Figure 11 the convex envelopes cross, so it is impossible to cut a 6″ plank the length of the flitch; in Figure 12 the convex envelopes are separated, and a possible pair of cut lines to produce a 6″ board is shown. (In the interests of error rejection, we position the cut lines to divide any excess width equally between the top and bottom

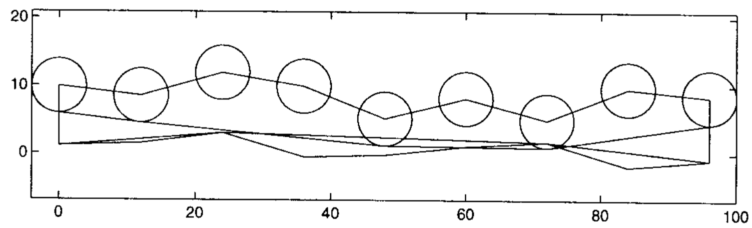


Figure 11: Convex chains overlap, so plank won't fit.

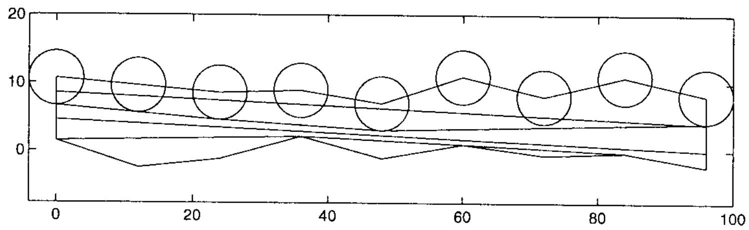


Figure 12: Successful separation and plank cutting.

edges, and to choose the midpoint of any nondegenerate interval in which the side slope of the cut board is allowed to lie.)

This method can be extended to determine the feasibility of templates more complicated than just a single plank. To accommodate a shorter board of width 4″ on top of the left end of the plank, it suffices to know the length of this board, and to increase by 4″ the radii of the circles in the corresponding portion of the upper flitch boundary. For this more general template, we cannot guarantee feasibility of any template that avoids the flitch vertices—one can imagine a straight-line segment of the boundary that cuts off the corner of the shorter board just mentioned—but we expect the slope of the separating line we find to be so small in practice that if we agree always to position both ends of any shorter boards directly above or below a vertex of the flitch, this problem will never become a practical issue. Indeed, an even coarser approximation may be adequate in practice: given enough data points, we may be able to enlarge the upper vertices not into disks, but simply into vertical segments, or to use segments at vertices near the flat parts of the template and disks near its nonconvex corners. (This will save some effort in the computation of the upper convex envelope for the augmented top edge, since each disk can in principle contribute a circular arc, whereas each segment will contribute just its lowest point.)

(c) Other Directions

The problem of finding how far along a given line segment one can travel before hitting the boundary of a given polygon in the plane or polyhedron in three-space is one that computer graphics experts need to solve in order to predict the shading of computer-generated solids. This is exactly the same problem we face, either in measuring distances inside the log from the butt end toward the top, or in seeking dimension lumber inside a given flitch. In the computer graphics community, the desired procedure is called "ray tracing." It would be instructive and likely very useful to explore the ray tracing literature and assess its possible impact on log optimization technology.

3 Rules of Thumb

So far we have dealt with quantitative approaches that could be used as components of a full cutting optimization system. If, instead, the goal is to make an educated (hopefully near-optimal) guess at a reasonable first cut without solving the full breakdown problem in advance, several ad-hoc procedures come to mind.

1 Different Algorithms in Parallel

In the absence of a perfect algorithm a selection of good and essentially different algorithms can help. If two algorithms exist, called A and B, and both are applied to every incoming log, one can choose the one producing the higher value for each instance. Even if algorithm A produces the better recovery 99 times for every 100 logs, the presence of algorithm B has made a contribution.

2 Hybridized Lookup and Optimization

As computers become cheaper and more powerful, there has been a move away from lookup tables and towards real-time optimization for calculating log breakdown strategies. A hybrid approach may be worth considering, however. Here the idea would be to single out a small list of essential log parameters that can be computed using the measurement information, and to use this list as a key into some large database of successful cutting patterns used in the past. In a pure lookup scheme, the database would provide the cutting axis directly; in a hybrid approach, the database would give intervals likely to contain the optimizing values of the continuous variables, and conventional optimization software would be instructed to restrict its search to these regions. This would either reduce execution time, or use the time available more efficiently, to make a more thorough exploration of the set of most interesting parameter values.

3 A Learning Approach

Another idea is to build a database that will become more intelligent the more it is used. Again, use essential log parameters to determine the "type" of each incoming log. Consult the database to find all of the cutting patterns which have applied to this type before. Apply them all to the log (under different orientations) and choose the best result, then cut the log according to this scheme. If the value recovery number that results falls below some preassigned efficiency level (which may depend on the log's type), send the image of the log to an offline computer that spends as much time as necessary to find what the best way to cut this log would have been. Then add this cutting scheme to the database.

The latter two approaches incorporate the notion of "essential parameters" that capture features of the log that are important when selecting a good cutting pattern and orientation. There are obvious suspects: the length, taper, and sweep of the whole log come immediately to mind, along with the sizes, eccentricities and principal axes orientations of the cross sections; other quantitative measures can also be imagined. (Characterizing the cross sections using the first few Fourier coefficients of their (2π-periodic) radius functions, for example, might provide an easy way to detect "cat face".) To identify correlations between the value recovery number and the parameters thought to be essential, we propose a statistical search[3] through idealized cutting reports. These would be based on the raw log data gathered by a particular mill in real operation, but with the actual breakdown replaced by a truly optimal cutting pattern generated by exhaustive offline search. If the observed correlations can capture the variation in optimal yield, then we can have some confidence that the parameters singled out as "essential" are sufficient. If not, and unexplained variation remains, it would indicate that some important property of the log has been overlooked; a natural response would be to look for it and repeat the statistical analysis described above. (Offline computation of the "idealized" cutting reports mentioned above may represent a significant computational challenge, but the payoff would be a database containing many patterns that improve on the best current practice. Running the statistical search on actual cutting schemes used in the real mill could also be done, but it would produce recommendations that are no better than what is done already ... although they may be produced faster than they are by full-scale optimization.)

Until the statistical analysis described above is done, we cannot be sure that a correlation between yield and a small number of log shape parameters will exist. If there turns out to be no correlation, we will be forced to conclude that there is no way to consistently produce good cutting patterns based on a small number of log parameters, and encouraged to pursue further the idea of efficient full-scale optimization in real time.

4 Acknowledgements

The authors thank Jan Aune of MacMillan Bloedel for proposing this problem, and for continued email correspondence during the weeks following the Industrial Problem Solving Workshop. Credit is also due to Jan Brdicko of MPM Engineering for coming to visit the group during the Workshop and providing useful information about the state of the log optimization industry.

5 Internet References

- Log data by anonymous ftp: `ftp.math.ubc.ca/pub/loew`
- Email address for Jan Aune: `je.aune@mbltd.com`
- Web page for MPM Engineering: `http://www.mpm-eng.com/`
- Web pages on Lumber Recovery from *Logging and Sawmilling Journal*: `http://www.forestnet.com/log&saw/lrf/lrf.htm`
- Web page for Nanoose Systems of Parksville (another maker of log optimization software), showing nice graphics of the optimization process: `http://www.nanoose.com/`
- Web page for Inovec Optimization and Control Systems of Eugene, Oregon (another maker of log optimization software): `http://www.inovec.com/index.shtml`

[3]The method of "Kriging", known to geophysicists, statisticians, and others who need to interpolate large data sets, may be of use here.